Universidade do Minho

Departamento de Informática

Programação Ciber-física

Trabalho Prático 1

Modelação e Análise de Sistemas Ciber-físicos

André Gonçalves PG46525
Francisco Peixoto PG47194

16 de maio de 2022

Parte 1

1. Decisões de design

O sistema modelado pode ser separado em quatro entidades que representam o funcionamento da junção em T. De forma a criar um modelo adequado para a representação da junção decidimos que era necessário representar os semáforos da estrada *major* e *minor*, a junção e o sensor na estrada *minor*.

O semáforo da estrada *minor* e os dois semáforos da estrada *major* são representados como dois processos separados. Os semáforos da estrada *major* têm as mesmas cores a qualquer momento por isso temos um único processo para os representar. As luzes dos semáforos da estrada *major* e *minor* têm de sincronizar de modo a não provocar acidentes de trânsito. Assim sendo pelo menos um dos semáforos tem que estar com luz vermelha.

O sensor da estrada *minor* deve inevitavelmente tornar o semáforo na estrada *minor* verde se carros são detetados.

A junção funciona como um objeto partilhado entre as estradas *major* e *minor*, a qual apenas uma pode ter direito, de modo aos carros da estrada passarem.

Outras decisões

- Ambos os semáforos podem estar com luz vermelho durante no máximo um segundo, até um deles se tornar verde;
- Os semáforos sincronizam de acordo com o sensor, nenhum dos semáforos pode passar de verde para vermelho e passar para verde outra vez antes de o outro ter passado para verde primeiro;
- O semáforo na estrada *major* encontra-se verde até ele estar verde durante mais de 30 segundos e o sensor ativar.

2. Modelos

O sistema tem 4 modelos: o **Sensor** para representar o sensor da estrada *minor*, **Road** que representa o recurso partilhado que é a junção, **MajorLight** e **MinorLight** que representam, respetivamente, os semáforos das estradas *major* e *minor*.

Sensor da estrada *minor*

O sensor comunica com ambos os semáforos. Ao detetar carros na estrada *minor* o sensor comunica com o semáforo *major* (`detected!`) para se tornar vermelho, de modo a libertar a junção para a estrada *minor*. O semáforo da estrada *minor* inevitavelmente torna-se verde durante exatamente 30 segundos e comunica com o sensor (`timeOut?`) e o sensor volta ao seu estado inicial.

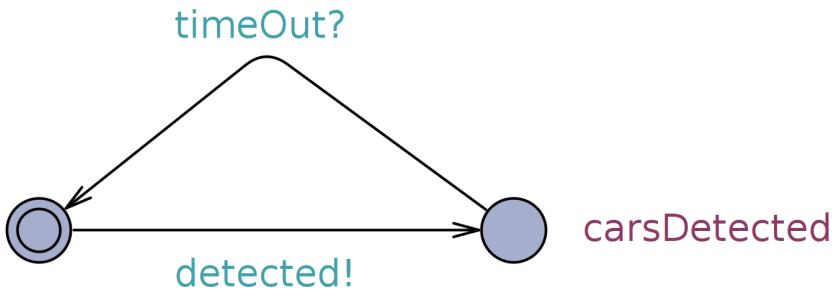

Figura 1: Sensor

Junção

Road representa o recurso partilhado da junção. Este força a sincronização dos semáforos, permitindo o acesso dos carros da estrada *major* e depois os da estrada *minor* ciclicamente. `majorStop?` informa

que o tráfego da estrada *major* parou e o tráfego da estrada *minor* pode avançar (`minorCross?`), e igualmente para `minorStop?` e `majorCross?`.

A variável `interval` mede o tempo em que o tráfego está parado, ou seja, tempo após um `majorStop` (`minorStop`) e `majorCross` (`minorCross`), não permitindo que o tempo sem tráfego seja menor que 1 segundo, o que implica implicitamente o tempo em que ambos os semáforos estão com luz vermelha.

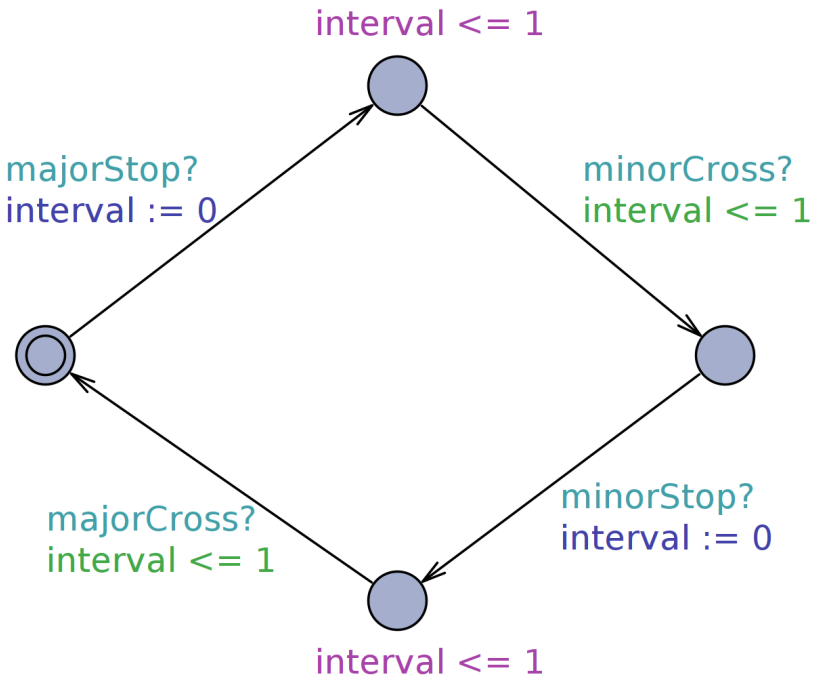

Figura 2: Road

Semáforos da estrada *major* - MajorLight

Os semáforos passam de verde, para amarelo e depois vermelho e retornam a verde. Estas cores são estados do modelo, entre cada um deles encontra-se um estado para medir o tempo de atraso entre trocas de luz.

O semáforo da estrada *major* encontra-se inicialmente com luz verde (estado `green`) e necessita de estar em `green` pelo menos 30 segundos antes de mudar de estado, que acontece quando o sensor ativa (`detected`). Depois ele permanece no estado intermédio durante 1 segundo. Após isto passa para amarelo (estado `yellow`) durante exatamente 5 segundos, após isto vai para o estado intermédio durante 1 segundo e depois para o estado vermelho indicando que o tráfego se encontra parado (ação `majorStop` com `Road`). Para o semáforo voltar a verde o tráfego da estrada `minor` tem de parar (uma vez que em `Road` apenas pode receber `majorCross` após receber `minorCross` e `minorStop`).

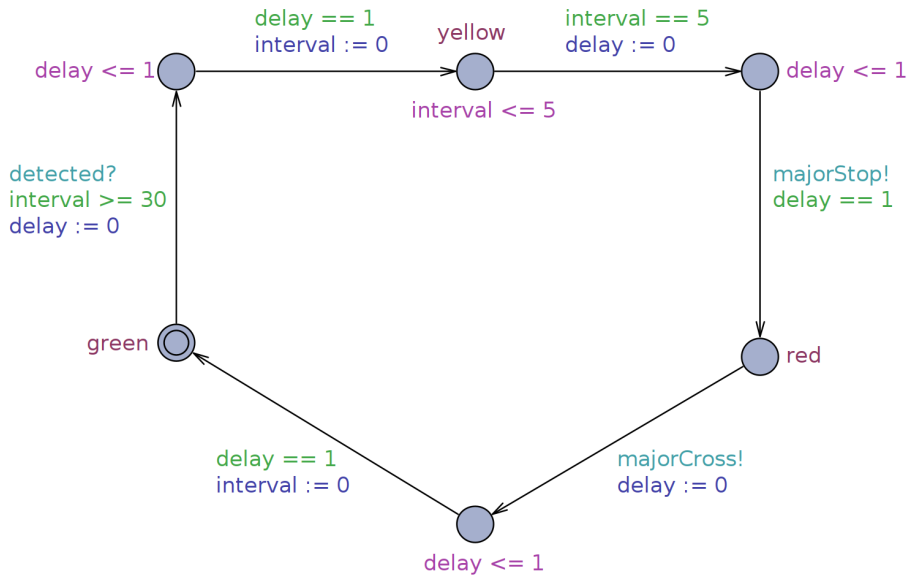

Figura 3: MajorLight

Semáforo da estrada *minor* - MinorLight

MinorLight é similar a **MajorLight**. Este semáforo encontra-se inicialmente a vermelho (estado `red`) e passa para verde quando o tráfego da estrada *major* para. Entre o estado `green` e o estado intermédio o semáforo envia `timeout!` ao sensor para indicar ao sensor que os 30 segundos em verde passaram e os modelos devem voltar aos estados iniciais.

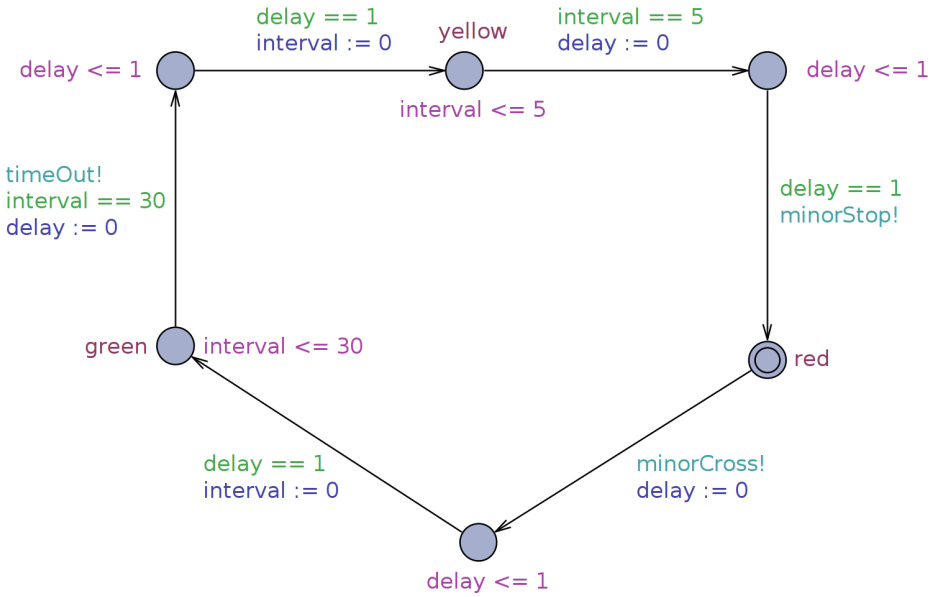

Figura 4: MinorLight

3. Fórmulas usadas para benchmarking

Todas as fórmulas dadas no enunciado são válidas no sistema. Mi e Ma são respetivamente os processos que representam o semáforo da estrada *minor* e os semáforos da estrada *major* e S é o processo do sensor.

1. $E\Diamond$ $Mi.green$ - o semáforo da estrada *minor* pode-se tornar verde;
2. $E\Diamond$ $Ma.red$ - o semáforo da estrada *major* pode-se tornar vermelho;

3. $A[]\ \neg deadlock$ - não há deadlock no sistema;
4. $A[]\ \neg(Mi.green \wedge Ma.green)$ - os semáforos não podem estar ambos com luz verde ao mesmo tempo;
5. $S.carsDetected --> Mi.green$ - se carros foram detetados pelo sensor então o semáforo da estrada *minor* torna-se inevitavelmente verde;
6. $A[]\ \neg((Mi.yellow \vee Mi.green) \wedge (Ma.green \vee Ma.yellow))$ - generalização do 4., os semáforos não podem estar ambos com luz verde ou amarela ao mesmo tempo, dado que isso poderia causar acidentes.

4. Conclusão

O sistema modelado representa adequadamente o cenário dado. As luzes de cada semáforo trocam de luzes da mesma forma que no mundo real e nunca existirá conflito no fluxo de tráfego entre as estradas *major* e *minor*. Se o sensor ativar então o semáforo da estrada *minor* torna-se inevitavelmente verde. Quanto aos requisitos dados no enunciado o sistema cumpre-os adequadamente.

Parte 2

1. Decisões de design

Neste sistema podemos ter N processos que cada representa um semáforo numa interseção, e cada um destes tem que ter um processo sensor respetivo ao estado do tráfego da sua estrada. Temos também um processo que representa o recurso partilhado da interseção das estradas para controlar o tráfego e um processo gestor que gere o fluxo do tráfego entre as estradas.

Cada sensor vai indicar ao gestor o estado do tráfego na sua estrada que pode ser não existente, baixo ou alto. O gestor contém duas filas uma para os semáforos que estão com tráfego baixo e outra para o tráfego alto. O gestor comunica com cada semáforo para informar que se pode tornar verde.

O gestor coloca os semáforos na fila de acordo com a sua intenção ou coloca-os numa lista de espera se já se tornou verde recentemente e assim espera que ambas as filas fiquem vazias para adicionar os semáforos às filas.

2. Modelos

Estrada - Road

Tal como na Parte 1 **Road** representa o recurso partilhado da junção, em que `lock` e `unlock` são referentes ao acesso das estradas à junção. Para um semáforo se tornar verde necessita de realizar `lock` com **Road** e ao voltar a vermelho liberta (`unlock`) a junção.

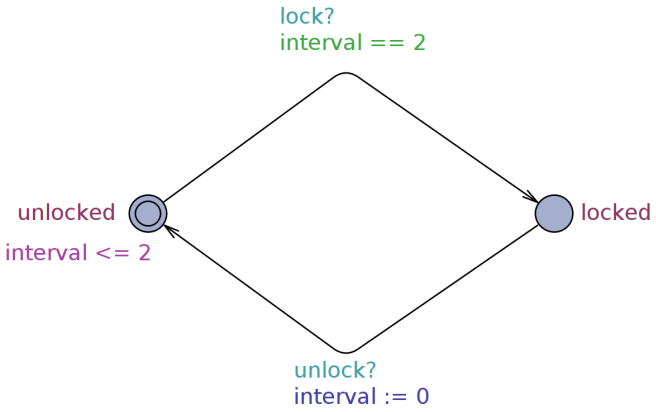

Figura 5: Road

Sensor

Cada semáforo possuí um sensor respetivo. Este sensor pode comunicar com o **Manager** para informar sobre o estado do tráfego, que pode ser baixo, alto ou não existente. Contudo depois de o **Sensor** comunicar com o **Manager** ele não pode comunicar outra vez antes de receber um `timout` do semáforo a informar que este já esteve verde. O sistema assegura que inevitavelmente se o sensor manda uma mesagem ao **Manager** o seu semáforo respetivo se tornará verde.

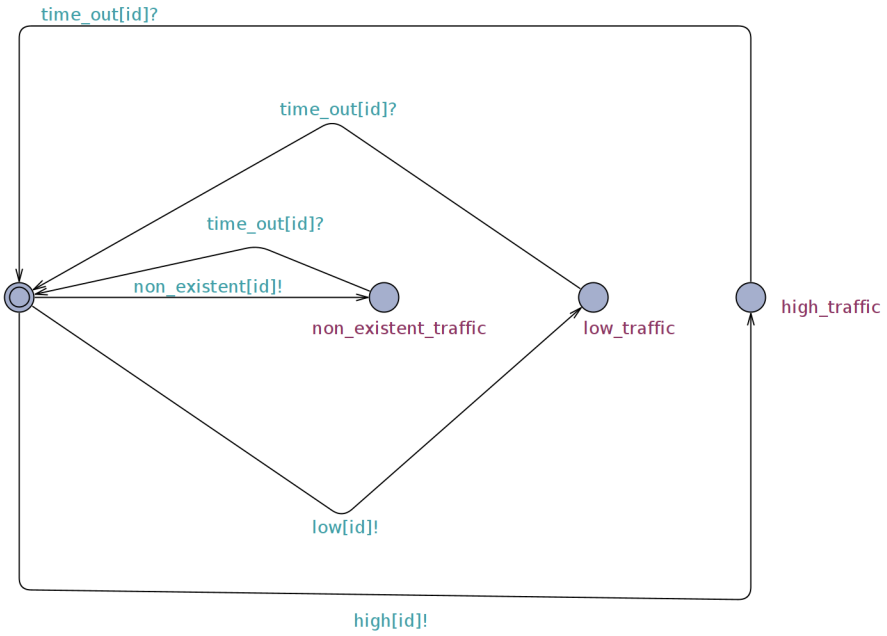

Figura 6: Sensor

Semáforo - Light

Todos os semáforos começam com luz vermelha e esperam que o **Manager** comunique com ele para informar para obter acesso à junção e se tornar verde. O tempo que o semáforo fica verde depende do estado do tráfego na sua estrada, dado pelo sensor e do estado do tráfego das outras estradas, se todos os outros tráfegos forem não existentes então o semáforo fica verde até outro sensor ativar. De modo a saber se o semáforo deve permanecer verde, este comunica com o **Manager** (`reset`), e o **Manager** devolve `reset_ok` se deve permanecer verde ou `reset_fail` se deve tornar-se vermelho e libertar a junção.

Ao passar de verde para amarelo o semáforo comunica com o sensor para este voltar a informar, caso necessário, sobre o estado atual da sua estrada.

Ao passar de amarelo para vermelho o semáforo liberta a junção e comunica com o **Manager** para o remover da fila atual e assim este pode informar outro semáforo na fila (ou o mesmo se não existe mais tráfego nas outras estradas e este ainda têm tráfego e comunicou com o **Manager**).

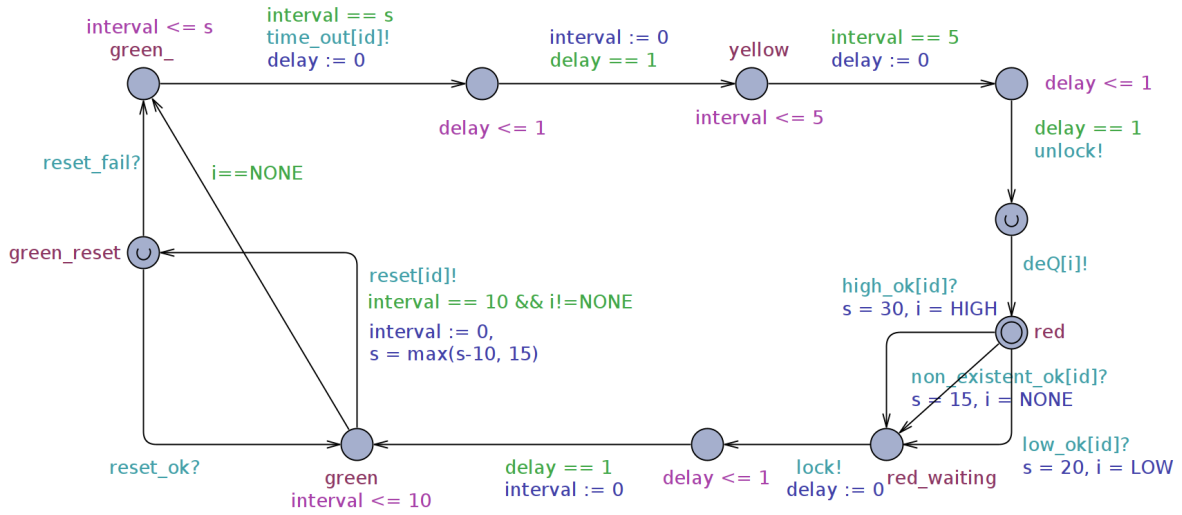

Figura 7: Light

Gestor de tráfego - Manager

O gestor de tráfego contém duas filas, uma para os sensores que detetaram baixo tráfego e outra para alto tráfego. Quando um sensor comunica com o gestor este adiciona o seu identificador a uma das listas. Caso nenhum sensor tenha comunicado com o gestor (tamanho das filas é igual a 0) um sensor pode comunicar `non_existent` com o gestor e depois o gestor vai permitir ao semáforo respetivo que ele se torne verde.

O gestor também contém uma lista de espera de forma a saber os semáforos que recentemente estiveram verde. Assim é garantido que quando um semáforo se torna verde, não se pode tornar verde até todos os outros que estão nas filas se tornem verde. Quando as filas estão vazias os semáforos que estavam na lista de espera são adicionados às filas respetivas.

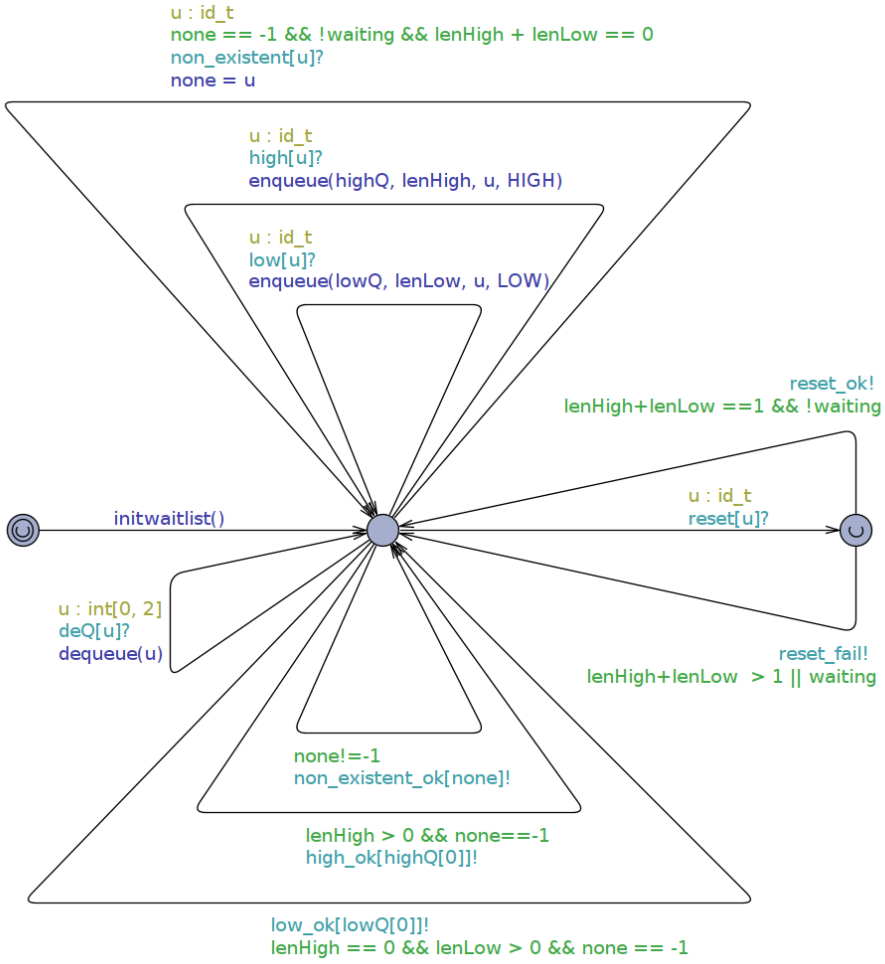

Figura 8: Manager

3. Fórmulas usadas para benchmarking

Utilisamos um sistema com três semáforos para testar as seguintes fórmulas, que foram verificadas pelo Uppaal:

1. $A[]\neg deadlock$ - não há deadlocks;
2. $(L0.red\_waiting) -- > L0.green$ - se um semáforo quiser passar de vermelho para verde inevitavelmente acontecerá;
3. $A[]\neg((L0.yellow \vee L0.green) \wedge (L1.green \vee L1.yellow) \wedge (L2.green \vee L2.yellow))$ - não pode acontecer de dois ou mais semáforos se encontrarem verde ou amarelo.

4. Conclusão

Outras melhorias poderiam ser feitas ao sistema relativas ao sensor uma vez que ele só pode comunicar com o gestor de uma vez, ou seja, ao comunicar o estado do tráfego ele tem que esperar que o semáforo faça um ciclo antes de comunicar de novo com o gestor e portanto pode acontecer de o sensor comunicar baixo tráfego ao gestor, mas imediatamente o tráfego aumentar e o sensor não pode informar imediatamente o gestor.

Na verificação das propriedades para o novo modelo, houve certas propriedades descartadas devido à mudança de contexto, visto que agora a mudança do sinal do semáforo depende da afluência do trânsito em cada estrada e não depende de regras estabelecidas para o cálculo de uma posição default.

Com o resto das propriedades apresentadas acima verificadas no sistema e considerando que o sistema modelado representa adequadamente o cenário dado, verificamos que os requisitos enunciados foram cumpridos.